Making the Move to DevOps in Highly Regulated Industries

Tech companies are rapidly adopting DevOps – the combined practice of IT and development operations – to improve product development and spur innovation. However, industrial and highly regulated companies have been slow to embrace DevOps, which is characterized by collaborative teams working with infrastructure that supports rapid-fire development. These companies are reluctant to make the move because it can be daunting to change entrenched legacy development programs, upfront investments need to be made in manpower and technologies, and there is often a significant lack of in-house expertise. This paper will look at how regulated companies can embrace DevOps practices and tools in order to benefit as much as their technically forward peers. Automated processes throughout the development cycle aid in quality of products, speed to release, and assurance of regulatory compliance. Gain insight into why some DevOps initiatives fail and how to avoid a similar fate, and learn how a phased approach allows for a gradual introduction of development infrastructure, practices, and processes that can be expanded to create a more nimble and responsive development organization.
Differentiating with DevOps

The practice of DevOps has steadily gained traction among companies that rely on software for business value, which is nearly every business in today’s fast-moving digital economy. More than a decade since its inception, DevOps is becoming a critical differentiator for companies that want to remain competitive.

Agile development, advanced cloud capabilities, and the need to incorporate security throughout the entire product lifecycle is impacting development organizations, making the culture of continuous learning embodied by DevOps more important than ever. Automation, agile, cloud and the efficiencies achieved by increased collaboration between IT and operations will continue to drive DevOps market growth to more than $12 billion by 2025 according to Grand View Research.

Organizations also are rethinking their development approaches in order to take advantage of emerging technologies. Artificial intelligence (AI), machine learning (ML), and the Internet of Things (IoT) are all inherently built on top of platforms that require an investment in DevOps practices to adequately manage and deploy devices that may reside in far-flung or harsh environments.

DevOps adoption is especially strong in technology firms, whose business models often rely on delivering software quickly to remain competitive. In fact, two leading state of DevOps reports — one from DevOps Research and Assessment (DORA) and another,1 published by Puppet, CircleCI, and Splunk found that among nearly four thousand development professionals surveyed, only a fraction came from industrials, manufacturing, government, energy, and medical industries, compared to nearly 40% from technology firms.

To understand why industrial and highly regulated industries are trailing tech firms in adoption, let’s first take a look at exactly what DevOps is and why it’s such a beneficial practice.

1 The Puppet 2019 State of DevOps Report requires registration for access.

DevOps Rate of Adoption by industry

<table>
<thead>
<tr>
<th>Industry</th>
<th>Rate of Adoption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>37%</td>
</tr>
<tr>
<td>Financial Services</td>
<td>13%</td>
</tr>
<tr>
<td>Retail/Consumer/eCommerce</td>
<td>8%</td>
</tr>
<tr>
<td>Other</td>
<td>7.5%</td>
</tr>
<tr>
<td>Industrial &amp; Manufacturing</td>
<td>5.5%</td>
</tr>
<tr>
<td>Life Sciences/Healthcare</td>
<td>5%</td>
</tr>
<tr>
<td>Government</td>
<td>5%</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>5%</td>
</tr>
<tr>
<td>Education</td>
<td>4.5%</td>
</tr>
<tr>
<td>Media/Entertainment</td>
<td>3.5%</td>
</tr>
<tr>
<td>Energy</td>
<td>3%</td>
</tr>
<tr>
<td>Insurance</td>
<td>2%</td>
</tr>
<tr>
<td>Non-profit</td>
<td>1%</td>
</tr>
</tbody>
</table>

*Accelerate: State of DevOps 2019, DORA
*2019 State of DevOps, Puppet, CircleCI & Splunk

DevOps Explained

The term DevOps is a deceptively simple word combining development and IT operations. Gartner provides an excellent definition:

“DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and it seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology — especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective.”

The practice, however, is a substantial shift away from the independent, siloed organizations common in many highly regulated companies to a collaborative, cross-functional way of developing products.
As product development teams strive to become more responsive to rapidly evolving business and customer needs, their infrastructure must support their ability to build, test, and deliver software faster and more quickly. DevOps is a cross-disciplinary, collaborative practice that allows organizations to deliver applications and services at high velocity, thus evolving and improving products at a faster pace than traditional siloed development, operations and QA organizations. This rapid pace of development allows organizations to become more competitive and better satisfy their customers.

DevOps implementations can take many forms, from large teams that bring together development (including product and QA teams) and operations (including system administrators and security professionals) under one umbrella, to smaller teams with experts that span the various disciplines. Successful implementations encompass the entire project lifecycle, from initial planning through production support.

DevOps is a set of practices and tools that facilitate rapid, self-directed development and automation to replace manual tasks. There is a focus on the technique of continuous integration (CI), where developers merge their code with the main code base on at least a daily basis to uncover integration issues and conflicts early in the process. From a broader development pipeline perspective, the ultimate goal of DevOps is to provide continuous delivery (CD) via fully automated build, test and deployment. The cultural shift and change in mindset from narrowly defined roles and skillsets to a collaborative, full lifecycle engagement is critical to the success of DevOps.

How Can DevOps Help Your Organization?

DevOps can help bridge gaps specific to the unique constraints of each organization by bringing modern toolsets and practices to existing environments. Companies with the highest levels of DevOps maturity enjoy both increased velocity of software delivery and increased stability. The DORA report reveals that these elite performers deploy code 208 times more often than those with low levels of maturity; recover from incidents 2,604 times faster; move from commit to deploy 106 times faster; and have a 7 times lower change failure rate.

Within industrial and highly regulated industries, automation supports ongoing compliance and reduces risk by providing traceability, clear audit trails and greater reliability. A well-designed DevOps platform takes advantage of today’s robust security technologies to improve the organization’s security posture. Further, a successful implementation can be scaled across other development organizations to extend these benefits throughout the company.

High Performing Vs. No Utilization

Teams with highly engaged DevOps practices experience large benefits compared to those who don’t utilize DevOps.

<table>
<thead>
<tr>
<th></th>
<th>High Performing</th>
<th>No Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>208x more frequent code deployments</td>
<td>208x</td>
<td></td>
</tr>
<tr>
<td>106x faster lead time from commit to deploy</td>
<td>106x</td>
<td></td>
</tr>
<tr>
<td>2,604x faster time to recover from incidents</td>
<td>2,604x</td>
<td></td>
</tr>
<tr>
<td>7x lower change failure rate</td>
<td>7x</td>
<td></td>
</tr>
</tbody>
</table>

*Accelerate: State of DevOps 2019, DORA  
*2019 State of DevOps, Puppet, CircleCI & Splunk
When DevOps Fails

In spite of the benefits DevOps offers, 75 percent of DevOps initiatives through 2022 will fail to meet expectations, according to Gartner. As DevOps implementations become more widespread, practitioners have learned common reasons why some projects aren’t successful and how to avoid the pitfalls that lead to failure.

Organizational changes are more likely than technology to be the culprit. It’s important to have leadership support across all impacted organizations to implement and encourage the collaborative effort required for a good DevOps program.

Unclear objectives including the lack of a well-understood business case and desired outcomes makes it difficult to make the appropriate organizational and technology changes needed for success. Selecting an achievable use case and ensuring all stakeholders understand the goals is critical for positive forward progress.

Unrealistic expectations that DevOps can solve numerous problems in a short timeframe will undermine program efforts. It should be clearly understood and communicated exactly what the team is working toward, with expectations that such significant changes will take time. Failures along the way are to be expected and embraced on the road to progress. The key is to set up infrastructure that makes failing fast – and recovering fast – the norm, so development progress can move forward with greater speed and reliability.

Insufficient skill levels, whether too few people with expertise in DevOps practices and tools, or a lack of DevOps knowledge, period, will make implementation difficult or impossible. This may necessitate bringing in outside talent to kickstart the program and provide guidance in hiring and training DevOps experts.

Lack of standardized tools and solutions can complicate efforts, especially if a tool is selected as the first step in a DevOps initiative with the misconception that a tool equates DevOps. By starting with thorough requirements work – rather than choosing technology first – the team can better understand exactly what tools and solutions are needed to work in conjunction with existing development environments, achieve the desired outcomes, and plan for expansion and scalability.

Why DevOps?

If you experience excessive downtime or lost productivity resulting from:

> Projects suffering from multiple points of failure
> Know-how residing with one or two key developers
> Frequent instances of “it works on my box” paradigm
> Regular production failures for unknown reasons
> Confusion about what codebase is deployed

DevOps can help eliminate these problems and provide company-wide benefits by:

> Reducing outage costs
> Spurring innovation
> Improving reliability, scalability, and business continuity
> Being able to respond to market demands and shifts more quickly
> Satisfying customers

Teams and individual contributors benefit from:

> Consistency of deployment practices and environments
> Expanding and distributing knowledge and skills
> Opportunity to innovate
> Improved productivity that leads to job satisfaction and reduces burnout
Regulated Industry Challenges

In addition to the pitfalls listed above, there are scenarios specific to highly regulated industries that can inhibit the adoption of DevOps practices. Many of these situations arose as a means to ensure higher levels of security and regulatory compliance but have evolved to become inefficient, monolithic applications that stymie innovation and don’t scale well. Companies find it difficult if not impossible to effectively respond to customer feedback and market forces to remain competitive – yet at the same time, they tend to remain highly risk-adverse.

Security is one of the most-cited reasons for companies within regulated industries to avoid doing DevOps. It is not uncommon for development projects in defense and other heavily guarded industries to institute physical air-gapped development environments that have no connections to outside networks. Even in these environments, there is significant value in establishing infrastructure and practices that apply modern tools, techniques, and practices. Advances in cloud security and the use of FedRAMP-compliant, government-specific cloud environments has opened the door to implementing new ways of working while keeping security at the forefront.

Regulatory compliance is on par with security as a barrier to DevOps adoption as organizations are reluctant to change their current compliance processes. However, using existing approaches, products or systems may be compliant at one point in time, but any changes in requirements, tools, or personnel can result in noncompliance that may not be discovered until a product is being readied for delivery. DevOps can help ensure ongoing compliance throughout the software development lifecycle (SDLC) through the use of automation, traceability and clear audit trails.

Segregation (or separation) of duties (SoD) is a third challenge that is required as part of certain regulations. It means that developers are not allowed to release code directly to production, so another person must be involved for every release. This layer of inefficiency contradicts the DevOps principles of rapid, frequent code releases, but can be overcome by keeping both parties within the DevOps organization.

The Importance of Culture

The significance of culture and organizational readiness cannot be understated. Before undertaking any DevOps efforts, it is critical that leaders within the organization understand and promote the cultural changes required for success. A culture of collaboration, continuous learning, and a willingness to try new things – with the support of leadership – is vital if the initiative is to succeed.

A Phased Approach to DevOps

Now that we’ve covered the challenges and benefits of employing DevOps, let’s take a look at how your organization can get started while minimizing disruption to existing development. A phased approach enables organizations to better understand the cultural, organizational and technology changes required for success. Pilot projects allow the program to be tested in a controlled environment and set the stage for a broader deployment.

Phase One: Architecture Analysis and Discovery

Phase One establishes a Proof of Concept (PoC) model that serves to begin cross-team collaboration, test variables, and identify gaps in expertise that will need to be addressed prior to broader deployment. Prior to creating the PoC, several activities need to occur:

> **Have an ultimate goal in mind.** Prior to setting out on any DevOps program, it is important to have a use case in mind; for example, automating build, test, and code release for a standalone application within a larger system. The PoC should be designed as a building block that supports the larger objective and demonstrates the benefit of DevOps to help garner support throughout the organization.

> **Identify the PoC team.** Establish a small team with knowledge and understanding of the objective, the current environment, and DevOps tools and principles. A can-do attitude...
and proven problem-solving skills are especially helpful.

> **Assess your current state.** Closely examine existing tooling, build, test, and delivery processes and make note of areas where inefficiencies occur. These could be things like onboarding new engineers where it takes an inordinate amount of time to locate documentation and configure their systems before they can start producing code, crucial source code residing on one engineer’s computer, or frequent test failures. Identify operational constraints that may include regulatory compliance, budgets, and available expertise. This process should not be rushed, as it may take weeks or months to fully understand the current environment.

> **Select tooling.** Depending on the results of the current state analysis, team members with DevOps expertise should select the appropriate tooling that will support a small-scale PoC as well as form the foundation for a broader rollout.

> **Build a PoC.** Keeping the overall goal in mind, build out a small PoC that addresses one or two areas from the assessment that can be improved. It is helpful to break down workflows into two areas. One is the local development environment. The second is continuous integration and continuous delivery (CI/CD), core principles of DevOps. CI refers to the ongoing integration of developer code into the main code repository and CD to the ongoing delivery of code across consistent development and production environments. The common thread across the two areas is to take an infrastructure as code (IaC) approach.

Popular technologies and tools include the use of Docker containers and a container platform orchestration tool like Kubernetes, with smart scripting and Vagrant to provision VMs for the local development environment so developers can be up and running quickly and environments can be deployed and updated easily. For CI/CD, Jenkins or GitLab’s CI can repurpose the containers or images created for the local environment so that there is a common set of tooling across both environments. A CI/CD server automates what may have been a highly manual process by creating a job every time a developer commits code that runs automated tests and produces an artifact with no manual intervention.

Ideally, the application of basic DevOps principles, practices, and tools during the PoC will yield positive results with a marked increase in efficiency and reliability as previously manual processes become automated.

---

**Standard DevOps Workflow**

- **SCM**: GitLab
- **Build**: Jenkins
- **Artifactory**
Phase Two: Implementation and Rollout

In Phase Two, the successful completion of the PoC establishes the grounds to move forward into production. Prior to this step, however, time and effort must be invested into additional planning.

> **Identify stakeholders.** Engage members from every organization that will be involved or impacted by the project, including product development, IT, QA, operations, marketing, finance, or other teams to secure agreement about the project scope and requirements.

> **Lay out requirements.** Spend time to clearly understanding the requirements of the project, from customer and market needs to regulations, financial considerations, manufacturing impacts, and more. Incorporate any requirements changes that may affect the new DevOps environment.

> **Add DevOps experts.** If there is a lack of DevOps expertise within the organization, it can be helpful to bring in outside talent to help move the program forward until in-house talent can be hired.

> **Move platform to production.** Once the appropriate planning and staffing have been completed, the DevOps platform infrastructure can be rolled out for wider adoption either on-premise, in the cloud, or a hybrid of the two. A container platform orchestration tool like Kubernetes is where DevOps infrastructure elements reside, such as specialized test environments, build environments, artifact repositories, security scanning tools, and more. These elements can be deployed relatively easily to the cluster and accessible to everyone as opposed to building environments on a one-off basis.

Phase Three: Optimization and Maintenance

Once the DevOps platform has been deployed and adopted by the organization, the last phase is to evolve the tool sets and enhance capabilities. The specifics will depend on each company’s development goals but are likely to include areas such as:

> **Security.** Well-designed DevOps environments are made up of technologies that incorporate advanced security features that require regular security updates. Security tools that meet the specific needs of the organization’s software and products can then be laid on top of it for robust security throughout.

> **New services.** Depending on the needs of the development organization, new services can be integrated and implemented on the new infrastructure.

> **Testing.** Test environments may evolve from unit test and integration test to deployment environments with hardware; for example, automated tests can be constructed around the software running on hardware and integrated into the entire development pipeline.

> **Scalability and performance optimization.** As the platform is rolled out to larger development projects or teams, adjustments may need to be made for scalability and performance optimization.

> **Emerging technologies.** As DevOps processes and tools streamline development and incorporate automation, emerging technologies such as AI and ML can be tapped to gain more awareness and initiate action, whether as part of testing, product performance and reliability, or user experiences.
Moving Ahead with DevOps

Although companies within highly regulated industries may have development environments that are more entrenched and restrictive than those in the technology space, the adoption of DevOps can deliver significant benefits that are likely to increase over time. By taking a phased approach to implementation, organizations can automate much of the development process which, in turn, enhances speed, reliability, and traceability. Compliance becomes easier to maintain even as requirements, personnel, tools, and products change.

The tools and technologies underpinning DevOps contain much higher levels of security than many legacy systems, and lend themselves to the addition of robust security tools and techniques that can be applied throughout the SDLC and to end products. Organizations who embrace DevOps can improve their responsiveness and ability to remain competitive, while creating an environment to spur innovation.

ABOUT THE AUTHOR

Matt Brewster is a DevOps Team Lead at Base2 Solutions. His proficiencies include deployment/build/test automation and infrastructure, systems integration and analysis, performance automation and analysis, infrastructure virtualization and containerization, aerospace information technology, software quality assurance and rapid prototyping. Matt’s DevOps/Infrastructure knowledge includes VMWare, AWS, Virtualbox, Docker, Vagrant, Ansible, Chef, Kubernetes, Linux, Nginx and Jenkins.

ABOUT BASE2

Base2 assists companies in giving form to digital innovation and delivers the processes and practices needed to develop complex interconnected systems. We help companies solve challenging problems by taking a whole systems approach to designing and developing the technology, processes, security and Systems Engineering required for success. Companies across heavily regulated industries, such as aerospace, defense, transportation and medical devices, have utilized our expertise and best practices to speed up product development and accelerate time to market. To learn more about how Base2 helps drive digital innovation, please visit www.base2s.com.